Needs analysis phase of a Cooperative Information System using viewpoints

Kahina Kessi, Zaia Alimazighi, Mourad Chabane Oussalah

Needs analysis phase of a Cooperative Information System using viewpoints

Kahina Kessi
USTHB University
LSI laboratory
Algeria
kkessi@usthb.dz

Zaia Alimazighi
USTHB University
LSI laboratory
Algeria
zalimazighi@usthb.dz

Mourad Oussalah
Nantes University
LINA Laboratory
France
mourad.oussalah@univ-nantes.fr

Abstract

In this paper we propose an approach which allows to define the Requirement Engineering phase of a Cooperative Information System (CIS). We used a notion of software engineering: the viewpoints. CIS is a complex system, it involves the cooperation of many stakeholders in a common purpose and each with their own viewpoint. This is why we used the notion of viewpoints, in order to decompose and partition the needs of a CIS according to the viewpoint of each stakeholder, to simplify its modeling. This approach defines a meta-model of viewpoint that will allow us to instance the necessary viewpoints to define the needs and requirements of a CIS.

1. Introduction

In a context of globalization of markets, companies of today face many challenges posed by: increased competition, the exceptional growth of services they must offer to their customers, the increased need to provide better quality of service and the necessity of cooperation and collaboration with other business partners to stay competitive in their activity domain and operating in socio-economic environments characterized by dynamism and increased turbulence. Thus, a company can no longer be content with a closed information system, which would cause it to be unable to exchange information with its collaborators and partners. The CIS area allows supporting the inter-company relations in order to improve interactions and communication between partners. In [9], the authors recognize that enterprise interoperability passes through the interoperability of their information systems to improve overall efficiency of a collaborative network. Size and complexity of these cooperative information systems is therefore growing, their complexity makes their design more difficult. It is so very important to understand the needs and requirements of the system, which leads us to be interested at the first phase of development of a CIS, i.e. the system analysis phase. What we want is to propose an approach which solves the problems related to the development of a CIS in order to propose a tool which let to support this approach using a notion of software engineering: the viewpoints. This article is organized as follow. In the second part we present our motivations and our interest regarding the proposal for an approach oriented viewpoint to the needs analysis phase of CIS. In the third part we present some methods that have addressed the same problem, before moving on to the fourth part where we present our VpCIs approach. Finally we conclude with our perspectives.

2. Motivation

The size and complexity of these cooperative information systems is therefore growing, their complexity makes their design more difficult. It is so very important to understand the needs and requirements of the system, which leads us to interest and define the first step of development of a CIS, ie the step of Requirements Engineering (RE). RE is the basis phase of the life cycle development of every project, defining what the stakeholders (users, customers, suppliers, developers, businesses) in a potential new system need from it, and also what the system must do in order to satisfy that need [6]. We need to find a set of requirements that reflect the needs of these stakeholders.
The methods that exist in the domain of RE in software engineering does not allow to address the complex needs of a CIS which involves the cooperation of many stakeholders in a common purpose and each with their own viewpoint. The use of existing approaches based on concepts of: goal, scenario or viewpoint have shown their limits, and work has been done for their integration into a single approach. [10] for example, proposed an approach named CREWS for requirements elicitation in which the authors use both goal and scenario, [2] proposed an approach inspired from CREWS where they integrated goal, scenario and viewpoints. there is also for example the methods VORD or

VOSE which used the viewpoints for the requirement engineering. But these methods have not highlighted the concepts of actors, team (group of actors), task, actions and interactions between actors that must be in a cooperative information systems. For the analysis needs phase of a cooperative information system other factors should be considered. It will be necessary to determine who does what, on what, when, after what and before what, we must define the systems actors and relations or actions and tasks that may exist between them. Which implies the intervention of different stakeholders (the expert, designer of CIS, domain user ...) involving several levels of modeling and multiple domains (generic domain (CIS), business application ...). We will try to solve these problems by proposing an approach that highlights these concepts using viewpoints with 5 dimensions in order to have a tool that allows to have a common formalism for each concept and remedy this complexity. This will allow us to decompose the needs of a CIS according to viewpoints of each stakeholder.
A viewpoint-based approach to requirements engineering recognizes that all information about the system requirements cannot be discovered by considering the system from a single perspective. Rather, we need to collect and organize requirements from a number of different viewpoints. A viewpoint is an encapsulation of partial information about a systems requirements. Information from different viewpoints must be integrated to form the final system specification [12]. There is several viewpoints methods in RE like : VOSE [5], [8], VORD [7], [11], [4], PreView [13], ISO/IEC/IEEE 42010 standard [1]. In addition, the main arguments in favor of an approach based on viewpoints in requirements engineering are:
- Systems usage is heterogeneous there is no such thing as a typical user. Viewpoints may organize system requirements from different classes of system end-user and other system stakeholders.
- Different types of information are needed to specify systems including information about the application domain, information about the systems environment and engineering information about the systems development. Viewpoints may be used to collect and classify this information.
- Viewpoints may be used as a means of structuring the process of requirements elicitation. [12]
The viewpoint establishes the conventions for constructing, interpreting and analyzing the view to address concerns framed by that viewpoint. Viewpoint conventions can include languages, notations, model kinds, design rules, and/or modeling methods, analysis techniques and other operations on views. A viewpoint is a way of looking at systems. [1]

In this approach, we interest to the cooperative information systems, we first propose a meta-model of viewpoint in the modeling level M2 in order to have a common formalism and to after instantiate the necessary viewpoints to describe a CIS in M1 level, we expect thereafter to provide a tool which allows to support this approach.

3. Related work and comparison

Several methods have proposed approaches to respond to needs and requirements of systems, we have oriented our choice to tooled methods as we later, want to propose a tool that will support our approach, knowing that we are interested in CIS and viewpoints, we then selected CREWS, Tropos-i* and MAMIE methods which are interested in collaborative work and we choose methods oriented viewpoint: VORD and VOSE. In the analysis needs phase of a cooperative information system we have to find these notions: Actor, team (group of actors), interaction between the actors, task (set of actions of a group of actors), action of an actor and requirements. we base our comparisons on these criteria.

3.1. The CREWS method

The CREWS [10] method (CREWS for Cooperative Requirements Engineering With Scenario), is developed under the ESPRIT project (European Research Project Reactive) approach.
The method has demonstrated its effectiveness in the requirements specification of a cooperative process by highlighting the concept of goal and scenario, the scenario has to make less fuzzy and more realistic goal. But what we can say about this method is that:
- The method does not highlights the actors, group of actors (team) and their descriptions or descriptions of the tasks of the system.
- Fragmentation of requirements captured in multiple scenarios makes difficult the assurance of completeness of the requirements specification.
- It also makes difficult to research the different aspects of the same system functionality across multiple scenarios.
- Sometimes non-functional concerns such as security and performance are treated secondarily compared to the central question of functional concerns.

3.2. The MAMIE method

MAMIE (from MAcro to MIcro level requirements Elicitation) [2] developed by bendjenna 2010, for the elicitation of requirements of an inter-organizational system.
The method drew inspiration from the CREWS method by coupling the goal (i*-method) and scenario (UML diagram) and added viewpoints (Preview) to describe non-functional requirements for the elicitation of requirements for a cooperative information system. But this method does not ex-

plicitly describes the relations between the actors and did not include the concept of task and team which are important in a cooperative information system.

3.3. Tropos-i* method

Tropos [3], which is requirements-driven in the sense that it is based on concepts used during early requirements analysis.
Goal modeling like Tropos-i* is an effective way to identify requirements. The argument of goal driven approaches is that the rationale for developing a system is to be found outside the system itself, in the enterprise in which the system shall function.
But it is difficult for domain experts to deal with the fuzzy concept of a goal. Yet, domain experts need to discover the goals of real systems. It is often assumed that systems are constructed with some goals in mind. However, goals are not given and therefore the question as to where they originate from acquires importance.
-The goal process do not reflect the actual situation but an idealized environmental one. Therefore, proceeding from this may lead to ineffective requirements. Thus, goal discovery is rarely an easy task. Add to this The method does not highlights the teams (group of actors) and their descriptions or descriptions of the tasks and groups actions of a system that we need to find a CIS.

3.4. VORD method

VORD [7] was developed at the University Lancasteret proposed by Sommerville and Kotonya (1996). It is a method for pre-validation of specifications, mainly dedicated to interactive systems and resolution of viewpoints.
The VORD method is useful in the detection of user needs, and also in identifying services that the user expects the system.
- However VORD method does not explicitly support the analysis of interactions between and within all viewpoints, it is oriented service, and does not represent the cooperation between actors, or the tasks descriptions and actions of the system, we don't find the notion of team which is important in a CIS.
- The viewpoints in VORD are predefined and rigid, it is not flexible and do not let the opportunity for developers to use their own notations and does not support changing requirements.

3.5. VOSE method

ViewPoint Oriented System Engineering, VOSE, was developed to "Imperial College" in London in the early 90s by Finkelstein, Nuseibeh et al. (1992) as a framework for the integration of development methods in compound systems (composite systems).
This method can well describe the requirements of composite systems but is not suited to cooperative information systems because it does not allow to define sets of interaction and cooperation between actors and we don't find the notion of team which is important in a CIS.
- As in VORD, the viewpoints in VOSE are predefined and rigid, it is not flexible and do not let the opportunity for developers to use their own notations.
The problem which we have found with these more structured approaches (VOSE, VORD, MAMIE) is that they are too rigid. They are based around the idea of a single type of viewpoint and require the specification to be fitted around this concept, this is why we propose our approach VpCIS which is flexible and explicit defining actors, interaction between the actors, teams, tasks and actions of the system, it let the developers and the users to define their needs and requirements using the representation style they want.

4. VpCIS approach

To minimize the disadvantage identified above, we propose a method based on an approach with 5 dimensions: **process, level of modeling, level of description, domain and expression mode** that integrates viewpoints from the needs analysis phase to describe the needs of a CIS, these viewpoints will allow us to describe the actors, interaction between them, tasks and actions, notions that we have find in a CIS. (an action is accomplished by an actor, a task is a group of actions assigned to a group of actors that have to work together to accomplish the task).
In what follows we present our approach VpCIs which consist at modeling level M2 in a language of viewpoint for a CIS: a viewpoint meta-model to describe actors of a CIS, this meta-model aims to describe an actor and a group of actors (team) using the viewpoints associated to the five dimensions that will allow us to describe the actor and his interactions with other actors and his tasks and actions. This meta-model will allow us to instantiate viewpoints actors and group of actors (team) at M1 modeling level and use these viewpoints at M0 level to describe a CIS in the real world.
We use as a mode of representation of the CIS the use case diagram, an UML diagram as UML provides tools that allow to describe the phases of analysis, design ....
The use case diagram allows us to highlight the actors, interactions between them and the tasks and actions, it allowed us to deduce two types of actors and actions in the meta-model: primary(intervene directly in the CIS) and secondary (intervene indirectly in the CIS).

A viewpoint VpCIs meta-model in the M2 level can be a simple actor or a group of actors as shown in Figure 1, it is

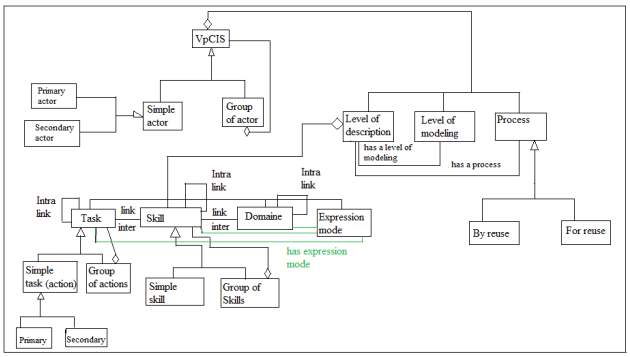

Figure 1. Meta-model of a viewpoint VpCIS

composed of the following:
1. Process: the construction process can be by or for reuse. To describe a CIS we can reuse actors and activities already defined in different cooperating systems or we can have to define new actors and activities.
2. Level modeling: the viewpoint VpCIS may intervene in one of the three layers of the architecture modeling:
- M2, the meta-models such as UML;
- M1, the models such as the class diagram;
- M0, the real world (case study);
This will allow us to describe the different categories of actors involved in each level modeling.
3. Level of description: this level comprises the following:
- *Activity:* may be a single action or group of actions to be done by an actor or group of actors.
- *Skill:* can be a skill of single actor or a group of skills of a group of actors.
- *4. Domain:* could be a generic domain or an application domain:
* *Generic domain:* in our case it is therefore the CIS, using the viewpoints we describe the actors and activities involved in modeling level M2 and M1 (expert, designer CIS ...)
* *Application domain:* using also the VPs we describe the actors and activity level M0 (actors and activities of the case study)
- *5. Expression mode:* representation mode of activities, skills and domain.
The viewpoint VpCIS simple actor and group of actors (team) and their attributes are described at M1level modeling.
We pass now to M1 level, where we instantiate the viewpoint actor and group of actors (team) from the meta-model of level M2 using the 5 dimensions:
The viewpoint group of actors (team) has the following attributes shown in table 1:

The viewpoint simple actor has the following attributes shown in table 2:

The table 3 shows a comparison between the methods: As regards **the notion of actors**: VOSE and VORD methods define viewpoints to describe the actors, CREWS and Tropos-i * define actors but does not describe them,

Table 1. Viewpoint Group of actors

Attributes	Description
Process	If the group of actors defines new actors the process is for reuse; if it uses actors already defined the process is by reuse
Level modeling	Level modeling where the group of actors intervenes
Name	Name of the group of actors.
Goal	What have to do the group of actors.
Expression mode of the group of skills	Representation mode of the skills group of the actors like an activity diagram or textual.
Group of Skills	Description of the skills set of the actors
Expression mode of the domain	Representation mode of the domain of the group of actors
Domain	Description of the domain of the group of actors
Activity (group of actions)	Name of the activity that is assigned to the group of actors
Expression mode of the activity (the group of actions)	Representation mode of the group of actions
Group of actions	Sets of actions of each actor who intervenes in the execution of the activity
Interaction between the actors	Inter link or intra link
Intra link	The actor has links with other actors from the same group of actors
Inter Link	The actor has links with other external actors from other groups of actors.
Required data: (functional requirements)	Requirement that can have the actor and information needed to accomplish the activity
Constraints on data: non-functional requirements	non-functional requirements that can have the actor.
Provided data	output information provided from the VP accomplishing his goal.
History	When start the activity and when finish.

Table 2. Viewpoint Simple actor

Attributes	Description
Process	If we define a new actor the process is for reuse; if it is already defined the process is by reuse
Level modeling	Level modeling where the actor intervenes
Type	Primary or Secondary
Name	Name of the actor
Goal	What have to do the actor
Expression mode of the skill	Representation mode of the actor skill like an activity diagram or textual
Skill	Description of the actor skill
Expression mode of the domain	Representation mode of the actor domain
Domain	Description of the actor domain
Action	Name of the actor action
Activity (group of actions)	Name of the Activity that is assigned to the actor
Expression mode of the action	Representation mode of the actor action
Action	Description of the actor action
Interaction between the actors	Inter link or intra link
Intra link	The actor has links with other actors from the same organization.
Inter Link	The actor has links with other external actors from other organizations.
Required data: (functional requirements)	Requirement that can have the actor and information needed to accomplish his action
Constraints on data: non-functional requirements	non-functional requirements that can have the actor.
Provided data	output information provided from the VP accomplishing his goal.
History	When start the action and when finish.

MAMIE defines the actors using a form.
The notion of team (group of actors): none of these methods described the concept of team.
Interaction between the actors: VOSE does not describe interaction between actors, in the other methods it is not explicit.
Activity (group of actions): none of these methods described the concept of activity.
Action: the action definition in the methods is not explicit.
Requirements : All the methods define the requirements.

5 Conclusion

In this paper we discussed our motivations for the use of viewpoints in the RE phase of a CIS. We proposed after, the VpCIS approach which defines a meta-model of viewpoints for the analysis needs phase of a CIS, to decompose the system needs and simplify the CIS modeling. We instantiated after that the viewpoints from the meta-model. We expect to develop a tool which allows to support this approach.

References

[1] International standard iso/iec/ieee 42010, systems and software engineering architecture description. *Systems and software engineering Architecture description*, 2011.

[2] H. Bendjenna. *Ingenierie des exigences pour les processus inter-organisationnels*. PhD thesis, University of Constantine and Toulouse, Algeria, France, 2010.

[3] J. Castro, P. Giorgini, M. Kolp, and J. Mylopoulos. Tropos: A requirements-driven methodology for agent-oriented software. 2004.

[4] M. Faisal and M. Hussain. Emphasizing requirement elicitation process for electronic payment secured system using vord methodology-a practical approach. *International Journal of Computer Science and Information Technologies*, 2012.

[5] A. Finkelsetin, B. J.Kramer, L. Nuseibeh, and M. Finkelstein. Viewpoints: A framework for integrating multiple perspectives in system development. *International Journal of Software Engineering and Knowledge Engineering*, 1992.

[6] E. Hull, K. Jackson, and J. Dick. *Requirements engineering*. Springer, 2010.

[7] G. Kotonya and I. Sommerville. Requirements engineering : Process and techniques. *John Wiley and Son, Great Britain*, 1998.

[8] S. Mehrdad, A. Finkelsetin, and G. Michael. Viewpoints. In *encyclopedia of software engineering. P. Laplante*, New York, 2010. Taylor and Francis.

[9] J. Monateri and M. Sapina. Dynamique des relations entre entreprises, strategies manufacturieres et arrangements contractuels durables. 1999.

[10] C. Rolland, C. Souveyet, and C. Ben Achour. Guiding goal modelling using scenarios. *IEEE Transactions on Software Engineering, Special Issue on Scenario Management*, 1998.

[11] A. Salem. Requirements analysis through viewpoints oriented requirements model (vord). *International Journal of Advanced Computer Science and Applications*, 2010.

[12] I. Sommerville and P. Sawyer. Viewpoints: Principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, 1997.

[13] I. Sommerville, P. Sawyer, and S. Viller. Viewpoints for requirements elicitation: a practical approach. *IEEE International Conference on Requirements Engineering*, 1998.

Table 3. Comparison between methods oriented viewpoints

Attributes	VORD	VOSE	CREWS	Tropos-i*	MAMIE	VpCIS
Process	No	No	No	No	No	yes
level modeling	Not explicit the method is in level M1	Not explicit the method is in level M1	Not explicit the method is in level M1	Not explicit the method is in level M1	Not explicit the method is in level M1	yes, we propose a meta-model in level M2 to instance Viewpoint in M1
level of description	No but it is implicit method for needs analysis	No but it is implicit method for needs analysis	No but it is implicit method for needs analysis	No but it is implicit method for needs analysis	No but it is implicit method for needs analysis	yes for needs analysis
Domain	yes	No	No		No	yes
System expression mode	Not explicit scenario	Not explicit graphic	Not explicit scenario	Not explicit graphic	Not explicit scenario	use case diagram
description formalism of a team (group of actors)	No	No	No	No	No	yes, viewpoint VpCIS
description formalism of an actor	VORD viewpoint	VOSE viewpoint	Not explicit	Not explicit	Not explicit form	yes, viewpoint VpCIS
actor type	yes	no	no	no	no	yes
Skills expression mode	not explicit: event scenario	not explicit: domain	not explicit scenario	not explicit goals	not explicit scenario	yes the user can use the mode he want
interaction between the actors the actors	No but it was proposed in the article [11]	no	not explicit we deduce it in the scenario	not explicit we deduce it in the graph	not explicit we deduce it in the scenario	yes we have inter link and intra link
Notion of activity	No	No	No		No	yes
Action	not explicit	not explicit	not explicit	not explicit	not explicit	yes
Required data: (functional requirements)	yes	yes	yes	yes	yes	yes
Constraints on data: non-functional requirements	yes	no	limited	limited	yes	yes
Provided data	No	No	No	No	No	Yes
Name	yes	yes	yes	yes	yes	yes
identification of goals	yes	yes	yes	yes	yes	yes
History	yes	yes	no	no	yes	yes
flexible viewpoint model	no	no	don't use viewpoint	don't use viewpoint	no	yes